niminst User's manual 1.3.5

Andreas Rumpf

August 21, 2020

Contents

1 Introduction

niminst is a tool to generate an installer for a Nim program. Currently it can create an installer for Windows via Inno Setup as well as installation/deinstallation scripts for UNIX. Later versions will support Linux' package management systems.

niminst works by reading a configuration file that contains all the information that it needs to generate an installer for the different operating systems.

2 Configuration file

niminst uses the Nim parsecfg module to parse the configuration file. Here's an example of how the syntax looks like:

```
# This is a comment.
; this too.

[Common]
cc=gcc     # '=' and ':' are the same
--foo="bar"   # '--cc' and 'cc' are the same, 'bar' and '"bar"' are the same (except for '#')
macrosym: "#"  # Note that '#' is interpreted as a comment without the quotation
--verbose

[Windows]
isConsoleApplication=False ; another comment

[Posix]
isConsoleApplication=True

key1: "in this string backslash escapes are interpreted\n"
key2: r"in this string not"
key3: """triple quotes strings
are also supported. They may span
multiple lines."""

--"long option with spaces": r"c:\myfiles\test.txt"
```

The value of a key-value pair can reference user-defined variables via the `$variable` notation: They can be defined in the command line with the `-var:name=value` switch. This is useful to not hard-coding the program's version number into the configuration file, for instance.

It follows a description of each possible section and how it affects the generated installers.

2.1 Project section

The project section gathers general information about your project. It must contain the following key-value pairs:

2.2 `files` key

Many sections support the `files` key. Listed filenames can be separated by semicolon or the `files` key can be repeated. Wildcards in filenames are supported. If it is a directory name, all files in the directory are used:

```
[Config]
Files: "configDir"
Files: "otherconfig/*.conf;otherconfig/*.cfg"
```

2.3 Config section

The `config` section currently only supports the `files` key. Listed files will be installed into the OS's configuration directory.

Key	description
Name	the project's name; this needs to be a single word
DisplayName	the project's long name; this can contain spaces. If not specified, this is the same as Name.
Version	the project's version
OS	the OSes to generate C code for; for example: "windows;linux;macosx"
CPU	the CPUs to generate C code for; for example: "i386;amd64;powerpc"
Authors	the project's authors
Description	the project's description
App	the application's type: "Console" or "GUI". If "Console", niminst generates a special batch file for Windows to open up the command line shell.
License	the filename of the application's license

Key	description
BinPath	paths to add to the Windows %PATH% environment variable. Example: BinPath: r"bin;dist\mingw\bin"
InnoSetup	boolean flag whether an Inno Setup installer should be generated for Windows. Example: InnoSetup: "Yes"

2.4 Documentation section

The documentation section supports the files key. Listed files will be installed into the OS's native documentation directory (which might be $appdir/doc).

There is a start key which determines whether the Windows installer generates a link to e.g. the index.html of your documentation.

2.5 Other section

The other section currently only supports the files key. Listed files will be installed into the application installation directory ($appdir).

2.6 Lib section

The lib section currently only supports the files key. Listed files will be installed into the OS's native library directory (which might be $appdir/lib).

2.7 Windows section

The windows section supports the files key for Windows specific files. Listed files will be installed into the application installation directory ($appdir).

Other possible options are:

2.8 UnixBin section

The UnixBin section currently only supports the files key. Listed files will be installed into the OS's native bin directory (e.g. /usr/local/bin). The exact location depends on the installation path the user specifies when running the install.sh script.

2.9 Unix section

Possible options are:

Key	description
`InstallScript`	boolean flag whether an installation shell script should be generated. Example: `InstallScript: "Yes"`
`UninstallScript`	boolean flag whether a deinstallation shell script should be generated. Example: `UninstallScript: "Yes"`

Key	description
`path`	Path to Inno Setup. Example: `path = r"c:\inno setup 5\iscc.exe"`
`flags`	Flags to pass to Inno Setup. Example: `flags = "/Q"`

2.10 InnoSetup section

Possible options are:

2.11 C_Compiler section

Possible options are:

3 Real world example

The installers for the Nim compiler itself are generated by niminst. Have a look at its configuration file:

```
; This config file holds configuration information about the Nim compiler
; and project.

[Project]
Name: "Nim"
Version: "$version"
Platforms: """
  windows: i386;amd64
  linux: i386;hppa;ia64;alpha;amd64;powerpc64;arm;sparc;sparc64;m68k;mips;mipsel;mips64;mips64el;powerpc;powerpc6
  macosx: i386;amd64;powerpc64
  solaris: i386;amd64;sparc;sparc64
  freebsd: i386;amd64;powerpc64;arm;arm64;riscv64;sparc64;mips;mipsel;mips64;mips64el;powerpc
  netbsd: i386;amd64
  openbsd: i386;amd64;arm;arm64
  dragonfly: i386;amd64
  haiku: i386;amd64
  android: i386;arm;arm64
  nintendoswitch: arm64
"""

Authors: "Andreas Rumpf"
Description: """This is the Nim Compiler. Nim is a new statically typed,
imperative programming language, that supports procedural, functional, object
oriented and generic programming styles while remaining simple and efficient.
A special feature that Nim inherited from Lisp is that Nim's abstract
syntax tree (AST) is part of the specification - this allows a powerful macro
system which can be used to create domain specific languages.

Nim is a compiled, garbage-collected systems programming language
which has an excellent productivity/performance ratio. Nim's design
```

Key	description
`path`	Path to the C compiler.
`flags`	Flags to pass to the C Compiler. Example: `flags = "-w"`

```
focuses on the 3E: efficiency, expressiveness, elegance (in the order of
priority)."""

App: Console
License: "copying.txt"

[Config]
Files: "config/*.cfg"
Files: "config/config.nims"

[Documentation]
; Files: "doc/*.html"
; Files: "doc/*.cfg"
; Files: "doc/*.pdf"
; Files: "doc/*.ini"
Files: "doc/html/overview.html"
Start: "doc/html/overview.html"


[Other]
Files: "copying.txt"
Files: "koch.nim"

Files: "icons/nim.ico"
Files: "icons/nim.rc"
Files: "icons/nim.res"
Files: "icons/nim_icon.o"
Files: "icons/koch.ico"
Files: "icons/koch.rc"
Files: "icons/koch.res"
Files: "icons/koch_icon.o"

Files: "compiler"
Files: "doc"
Files: "doc/html"
Files: "tools"
Files: "tools/nim-gdb.py"
Files: "nimpretty"
Files: "testament"
Files: "nimsuggest"
Files: "nimsuggest/tests/*.nim"

[Lib]
Files: "lib"

[Other]
Files: "examples"
Files: "dist/nimble"

Files: "tests"

[Windows]
Files: "bin/nim.exe"
Files: "bin/nimgrep.exe"
Files: "bin/nimsuggest.exe"
Files: "bin/nimble.exe"
Files: "bin/vccexe.exe"
Files: "bin/nimgrab.exe"
Files: "bin/nimpretty.exe"
Files: "bin/testament.exe"
Files: "bin/nim-gdb.bat"

Files: "koch.exe"
Files: "finish.exe"
; Files: "bin/downloader.exe"

; Files: "dist/mingw"
Files: r"tools\start.bat"
BinPath: r"bin;dist\mingw\bin;dist"
```

```
;           Section | dir | zipFile | size hint (in KB) | url | exe start menu entry
Download: r"Documentation|doc|docs.zip|13824|https://nim-lang.org/download/docs-${version}.zip|overview.html"
Download: r"C Compiler (MingW)|dist|mingw.zip|82944|https://nim-lang.org/download/${mingw}.zip"
Download: r"Support DLLs|bin|nim_dlls.zip|479|https://nim-lang.org/download/dlls.zip"
Download: r"Aporia Text Editor|dist|aporia.zip|97997|https://nim-lang.org/download/aporia-0.4.0.zip|aporia-0.4.0\
; for now only NSIS supports optional downloads

[WinBin]
Files: "bin/makelink.exe"
Files: "bin/7zG.exe"
Files: "bin/*.dll"

[UnixBin]
Files: "bin/nim"


[Unix]
InstallScript: "yes"
UninstallScript: "yes"
Files: "bin/nim-gdb"
Files: "bin/nim-gdb.bash"


[InnoSetup]
path = r"c:\Program Files (x86)\Inno Setup 5\iscc.exe"
flags = "/Q"

[NSIS]
flags = "/V0"

[C_Compiler]
path = r""
flags = "-w"


[deb]
buildDepends: "gcc (>= 4:4.3.2)"
pkgDepends: "gcc (>= 4:4.3.2)"
shortDesc: "The Nim Compiler"
licenses: "bin/nim,MIT;lib/*,MIT;"

[nimble]
pkgName: "compiler"
pkgFiles: "compiler/*;doc/basicopt.txt;doc/advopt.txt;doc/nimdoc.css"
```